

Claims 1 - 16 have been canceled.

Claim 17 (new)

A computer implemented method of presenting structured entity as a tree-like structure comprising the steps of:

providing functionality to display, manipulate, search, and store one or more tree-like structures having nodes, which represents one or more instance(s) of a structured entity;

where said structured entity presented in a tree-like form can represent pieces of code in a programming or scripting language or data, or binary file, or a communication protocol message;

basing said tree-like structure on rules based on grammar of the structured entity, the tree-like structure when representing piece(s) of code in a programming language or scripting language can support multiple programming languages and scripting languages in the same display; and

displaying data or binary file, mapping the content of the file onto the tree, wherein the tree-like structures can be selected to display in conventional state and back to tree-like structure.

Claim 18 (new)

The method of claim 17, further comprising the steps of:

performing validation and verification of the tree-like structures, enforcing conformance to rules based on grammar like BNF, EBNF or user defined extensions;

applying said rules to validate tree-like structure during creation and manipulation by a user;

creating an information piece by selecting a portion of a tree-like structure representing code in a programming or scripting language and create a standalone structure, grouping information pieces logically or semantically;

wherein said validation prevents consistency errors during modification of tree-like structures, build of tree-like structures by user according to user preferences and rules to valid tree-like structure during build; and

building said tree-like structures by the user according to rules and user preferences dynamic build and data of structure entity, a plurality of trees wherein each tree is coupled to its own programmable representation based on the structured entity type or content

Claim 19 (new)

The method of claim 18, further comprising the steps of:

providing display formats or structured entities as conventional state displays and/or tree-like structures and back, having the ability to drill down display of nodes in tree-like structures and

providing commonly accepted code indentation rules, syntax coloring and highlighting, hide portion of tree-like structure and display binary file in a tree-like structure.

Claim 20(new)

The method of claim 18, further comprising the steps of:

manipulating the tree-like structure based on rules, whereby re-factoring is provided by copy/replicate/change/deletion of a selected range of nodes and use of cut, paste, or drag & drop of nodes in tree, an ability to move a node or range of nodes in a tree-like structure, an ability to create standalone tree-like structures from existing tree-like structures, an ability to merge tree-like structures and ensure validity of structure with rules, modifications and merge operations provide notifications to user at the time of operation, grammar based rules used to determine which structures can be merged at the time of change and which can not at the time of operation and the ability to edit the content of the nodes.

Claim 21 (new)

The method of claim 20, further comprising the steps of:

creating new tree-like structures from existing instances and producing another instance of a representation of a structured entity that includes a subset of the nodes of the original structured entity; and

whereby said instance can be edited separately and can be merged back into the original tree based on changes parsed back to the tree using the rules to merge back into tree-like structure.

Claim 22(new)

The method of claim 18 further comprising the steps of:

searching, updating, and modifying contents of nodes in a tree-like structure, wherein updating of selected nodes can be performed with a prerecorded

sequence of actions which can be applied to a set of nodes of tree-like representation based on the factors common for the set of nodes or their content.

Claim 23 (new)

The method of claim 18, further comprising the steps of:

providing versions and locks of nodes and their related content in a tree-like structure, the structured entity representation; and

dynamically rebuilding according to the appropriate version of the rule representation and source control based on the tree-like representation when the content is locked for manipulations at the level of nodes or sets of nodes of the tree, so that multiple users can work on different parts of the content of the same structured entity simultaneously.

Claim 24 (new)

A computer implemented system having instructions stored on a computer readable medium when executed perform the operations of presenting a tree like structure comprising the steps of:

providing the ability to display, manipulate, search, and store one or more tree-like structures representing one or more instance(s) of a structured entity, where said structured entity presented in a tree-like form can represent pieces of code in a programming or scripting language or data, or binary file or a communications protocol message;

basing the tree-like structure on rules based on grammar of the structured entity, the tree-like structure when representing piece(s) of code in a programming language or scripting language can support multiple programming languages and scripting languages in the same display, wherein the tree-like structure(s) consist of nodes; and

mapping content onto the tree, wherein the tree-like structures can be selected to display in conventional state and back to tree-like structure when displaying data or binary file.

Claim 25 (new)

The computer implemented system of claim 24, further comprising:

performing validation and verification of the tree-like structures, enforcing conformance to rules based on grammar like BNF, EBNF or user defined extensions, said rules used to validate tree-like structure during creation and manipulation by a user;

providing the ability to create an information piece by selecting a portion of a tree-like structure representing code in a programming or scripting language and create a standalone structure;

grouping pieces of information logically or semantically;

building the tree-like structures by the user according to rules and user preferences dynamic build and data of structure entity, a plurality of trees wherein each tree is coupled to its own programmable representation based on the structured entity type or content, and

performing validation to prevent consistency errors during modifications of tree-like structures build of tree-like structures by user according to user preferences and rules to valid tree-like structure during build.

Claim 26 (new)

The computer implemented system of claim 25, further comprising

providing display formats of structured entities as conventional state displays and/or tree-like structures and back, the ability to drill down display of nodes in tree-like structure; and

provide commonly accepted code indentation rules, syntax coloring and highlighting, hide portion of tree-like structure and display binary file in a tree-like structure.

Claim 27 (new)

The computer implemented system claim 26, further comprising the:

manipulating the tree-like structure based on rules, whereby re-factoring is provided by copy/replicate/change/deletion of a selected range of nodes and use of cut/paste or drag & drop of nodes in tree, an ability to move a node or range of nodes in a tree-like structure, an ability to create standalone tree-like structures from existing tree-like structures, an ability to merge tree-like structures and ensure validity of structures with rules, modifications and merge operations provide notifications to user at the time of operation, grammar based rules used to determine which structures can be merged at the time of change and which can not at the time of operation and the ability to edit the content of the nodes.

Claim 28 (new)

The computer implemented system of claim 26, further comprising the:

creation of new tree-like structures from existing producing another instance of a representation of a structured entity that includes a subset of the nodes of the original structured entity; and

editing separately the instance and allowing the instance to be merged back into the original tree based on changes parsed back to the tree using the rules to merge back into tree-like structure.

Claim 29 (new)

The computer implemented system of claim 25, further comprising the:

performing a search, update and modify the contents of nodes in a tree-like structure, update of selected nodes can be performed with a prerecorded sequence of actions can be applied to a set of nodes of tree-like representation based on the factors common for the set of nodes or their content.

Claim 30 (new)

The computer implemented system of claim 25, further comprising the:

providing versions and locks of nodes and their content in a tree-like structure, the structured entity representation can be dynamically rebuilt according to the appropriate version of the rule representation and source control based on the tree-like representation when the content is locked for manipulations at the level of nodes or sets of nodes of the tree, so that multiple users can work on different parts of the content of the same structured entity simultaneously.

Claim 31 (new)

A computer comprising:

a processor,

a memory,

a display,

the processor executing instructions encoded in the memory as a data structure that through the execution of the instructions by an application presents a tree-like data structure in the memory which includes interoperability to provide

interrelations between the processor executing the instructions of the applications and the data structures in the memory comprising:

processor displaying on the display, manipulating in the memory, searching in the memory, and storing in the memory one or more the tree-like structures representing one or more instance(s) of a structured entity,

processor storing in memory the structured entity presented in a tree-like form can represent pieces of code in a programming or scripting language or data, or binary file or a communications protocol message; said tree-like structure is based on rules based on grammar of the structured entity,

processor storing in memory the tree-like structure when representing piece(s) of code in a programming language or scripting language can support multiple programming languages and scripting languages in the same display, the tree-like structure(s) consist of nodes, and

display presenting data or binary file wherein the content of the file is mapped onto the tree, the tree-like structures can be selected from memory to display, in conventional state and back to tree-like structure.

Claim 32 (new)

The computer of claim 31, further comprising:

processor performing validation and verification of the tree-like structures in memory, enforcing conformance to rules based on grammar like BNF, EBNF, or user defined extensions, said rules used to validate tree-like structure in memory during creation and manipulation by a user entries,

processor creates an information piece by selecting a portion of a tree-like structure representing code in a programming or scripting language and creates a standalone structure within the memory,

processor performing grouping information pieces logically or semantically within the memory,

processor building tree-like structures in memory by the user according to rules and user preferences dynamic build and data of structured entity, a plurality of trees wherein each tree is coupled to its own programmable representation based on the structured entity type of content; and

processor builds the tree-like structures by user according to user entered preferences and rules to validate tree-like structure during build within the memory; and

processor performing validation that prevents consistency errors during modification of tree-like structures in the memory

Claim 33 (new)

The computer of claim 32, further comprising:

display providing display formats of structured entities from the memory as conventional state displays and/or tree-like structures and back, the ability to drill down display of nodes in tree-like structure, provide commonly accepted code indentation rules, syntax coloring, and highlighting, hide portion of tree-like structure and display binary file in a tree-like structure in the memory.

Claim 34 (new)

The computer of 33, further comprising:

manipulating the tree-like structure based on rules, whereby re-factoring is provided by copy/replicate/change/deletion of a selected range of nodes and use of cut/paste or drag & drop of nodes in tree, an ability to move a node or range of nodes in a tree-like structure, an ability to create standalone tree-like structures from existing tree-like structures, an ability to merge tree-like structures and ensure validity of structure with rules, modifications and merge operations provide notifications to user at the time of operation, grammar based rules used to determine which structures can be merged at the time of change and which can not at the time of operation and the ability to edit the content of the nodes.

Claim 35 (new)

The computer of claim 33, further comprising:

processor interoperable with memory to create new tree-like structures from existing producing another instance of a representation of a structured entity that includes a subset of the nodes of the original structured entity, said instance can be edited separately and can be merged back into the original tree based on changes parsed back to the tree using the rules to merge back into tree-like structure.

Claim 36 (new)

The computer of claim 32, further comprising the:

processor interoperable with memory to search, update and modify contents of nodes in a tree-like structure, update of selected nodes can be performed with a prerecorded sequence of actions can be applied to a set of nodes of tree-like representation based on the factors common for the set of nodes or their content in memory

Claim 37 (new)

The computer of claim 32, further comprising:

processor interoperable with memory to implement versions and locks of nodes and their content in a tree-like structure in memory, wherein the structured entity representation can be dynamically rebuilt by the processor in memory according to the appropriate version of the rule representation and source control based on the tree-like representation when the content is looked for manipulations at the level of nodes or sets of nodes of the tree, so that multiple users can work on different parts of the content of the same structured entity simultaneously in memory.